[0047] It will be further appreciated that, in embodiments involving contour detection, the video image of the object being held by the user will be a projection of the 3D object into a 2D plane (i.e. the image plane), and so changes in orientation will need to be inferred from changes in appearance of the object in the 2D image. For asymmetric objects, such as bananas, it may be that a direction is defined based on properties of the object. For example, bananas usually have a narrower tip at one end, and so this part of the corresponding contour may be identified as corresponding to e.g. the north/south/east or west direction, with changes in orientation being detected relative to this axis.

[0048] In some examples, it may be that a central point of the object is identified using e.g. image moments, and the central point is used to define an origin of a coordinate system. A vector may be defined as extending from the origin of the coordinate system to a given point on the contour (e.g. the banana nib). Rotation of the object in the clockwise and anticlockwise directions (e.g. about the yaw-axis) may then be detected based on rotation of the vector relative to arbitrary pre-defined axes (e.g. x- and y-axes). Generally, it is expected that rotation in the clockwise and anticlockwise rotations will result in greater changes in direction of the corresponding vector, and so can be distinguished from rotation in the pitch and yaw axes. This rotation can then be translated into e.g. steering inputs.

[0049] In FIG. **2**, the object pose detector **203** is shown as receiving an input from the object detector **202**. The input received from the object detector **202** may correspond to the one or more contours identified as corresponding to the object by the object detector **202**.

[0050] FIG. **4** schematically shows an example of a contour detected for the banana shown in FIG. **3**. In FIG. **4**, the banana is shown as having contour **402**, and a corresponding linear representation **404** indicated by a dashed line. It can be seen in FIG. **4** that at least some of the outline corresponds to where the user's thumbs were when holding the banana. Nonetheless, in FIG. **4**, the largest contour still corresponds to the object that is being held, and can therefore be used to detect the object in the image.

[0051] Returning to FIG. **2**, the system **200** also comprises a user input generator **204** operable to generate a user input based on detected changes in the pose of the non-luminous object. The user input corresponds to a command for processing by an instance of a video game being executed at a video game playing device. The command may be transmitted to e.g. the CPU of a video game playing device, for processing thereat. The video game playing device may then update the display of a virtual character or object in accordance with the input generated by the user input generator **204**. Generally, the generated user inputs may be transmitted to a so-called video game unit, which comprises one or more processors (e.g. in the form of a CPU and GPU) for updating the state of a video game in accordance with received player inputs.

[0052] In embodiments where contour detection is used to detect and track the object, the user input generator **204** may be configured to generate different respective user inputs based on whether the orientation, position and area of the contour is detected as changing (relative to a default orientation, position and area of the contour) in the obtained images. For example, rotation of the contour (or representation thereof) may be used to generate a directional command corresponding to steering a moving object in a corresponding direction or changing a viewpoint of a virtual camera in a corresponding direction. In some examples, changes in area of the contour may be used to control a rate of travel in the forward and/or backward directions; for example, an increase in area may correspond to a 'travel forward' command, or 'accelerate' command; a decrease in an area may correspond to 'travel backward' command or 'brake' command. Moreover, changes in position of the contour (rather than rotation about an axis) may be used to generate a pause command. For example, movement of the contour above a threshold height, or out of the frame of the obtained images, may correspond to a pause operation.

[0053] It will be appreciated that the components of the above-described system **200** may be implemented at one device or distributed across multiple devices. In some examples, the video game playing device may comprise the input unit **201**, object detector **202**, object pose detector **203** and user input generator **204**. That is, the video game playing device may be configured to perform the functions of each component described above (i.e. each component may correspond to a respective module being executed by one or more processors at the video game playing device). In other examples, it may be that the input unit **201**, object detector **202** object pose detector **203** and user input generator **204** are split between an intermediate device that is communication with the video game playing device. It will be further appreciated that, in some examples, the system **200** may further comprise a camera for capturing the images of the user holding the non-luminous object.

Two-Object Controller

[0054] In some embodiments, a user may wish to use two non-luminous passive objects as a video game controller. The two non-luminous may be of the same type, e.g. a player may hold two bananas—one in each respective hand; or e.g two oranges—one in each respective hand. The above-described system **200** may be used to convert the movements of two objects being held by the user into video game inputs, as will be described below.

[0055] Referring to FIG. **2**, the input unit **201** may be configured to obtain images of at least two non-luminous objects being held by the user. Generally, each object will be in a respective hand of the user.

[0056] The object detector **202** may be configured to detect two non-luminous objects being by the user in the obtained images. In such examples, the object detector **202** may be configured to provide an input to object pose detector **203** and user input generator **204**, so as to indicate that a 'dual object' control mode is to be used for controlling the virtual object or character in the video game. In response to receiving this input, the object pose detector **203** and user input generator **204** may switch to a different (or adjusted) mode of object tracking and user input generation.

[0057] In some embodiments, the object detector **202** is configured to detect each object in the obtained image by detecting respective contours corresponding to the at least two non-luminous objects. As described previously, detecting contours of each object may involve filtering colours from the images known not to correspond to the objects being held by the user. As will be appreciated it may be more effective for the objects being held by the user to be the same colour (e.g. two bananas or two oranges), to simplify the filtering operation and subsequent identification of contours corresponding to those objects. As before, it may be that the